

CLAIMS

What is claimed is:

1. A method, comprising:
 - detecting creation of a thread, the thread running on a processor that is part of a distributed computing environment;
 - after detecting the creation of the thread, receiving a request from a requesting software to allocate a thread local storage associated with the thread;
 - scanning a data structure for a smallest suitable class size, the data structure including a list of thread local storage address size classes, each thread local storage address size class having a plurality of thread local storage addresses;
 - determining whether the smallest suitable size class is found;
 - if the smallest suitable size class is found, determining whether thread local storage of the smallest suitable size class is available;
 - if the smallest suitable size class is found, and if thread local storage of the smallest suitable size class is available, selecting a thread local storage address from among those thread local storage addresses belonging to the smallest suitable size class; and
 - if the smallest suitable size class is found, and if thread local storage of the smallest suitable size class is available, returning the thread local storage address to the requesting software.
2. The method of claim 1, wherein the data structure is resident in a private memory of each processor in a multiprocessor configuration.
3. The method of claim 1, further comprising: if the smallest suitable size class is not found,
 - scanning the data structure for a next larger suitable size class;
 - determining whether the next larger suitable size class has been found;
 - if the next larger suitable size class is found, selecting a thread local storage address

from the next larger suitable size class; and
if the next larger suitable size class is found, returning the thread local storage address
to the requesting software.

4. A method, comprising:

detecting destruction of a thread, the thread running on a processor that is part of a distributed computing environment;
after detecting the destruction of the thread, receiving a request from a requesting software to deallocate a thread local storage associated with the thread;
scanning a data structure for a smallest suitable size class, the data structure including a list of thread local storage address size classes, each thread local storage address size class having a plurality of thread local storage addresses;
determining whether the smallest suitable size class is found;
if the smallest suitable size class is found, determining whether thread local storage of the smallest suitable size class is available;
if the smallest suitable size class is found, and if thread local storage of the smallest suitable size class is available, creating a new entry of the smallest suitable size class;
if the smallest suitable size class is found, and if thread local storage of the smallest suitable size class is available, denoting the new entry in a thread local storage address of the smallest suitable size class; and
if the smallest suitable size class is found, and if thread local storage of the smallest suitable size class is available, inserting the new entry into the data structure.

5. The method of claim 4, further comprising deallocating the thread local storage without inserting the new entry into the data structure, if a smallest suitable size class is not found, or if thread local storage of the smallest suitable size class is not available.

6. A method, comprising:

receiving a request from a requesting thread for a thread local storage address, the thread local storage address belonging to an owning thread running on a processor that

- is part of a distributed computing environment;
searching a data structure for the thread local storage address using a code identifying
the owning thread; and
returning the thread local storage address, that belongs to the owning thread, to the
requesting thread.
7. The method of claim 6, wherein the data structure is resident in a shared memory unit.
8. An apparatus, comprising:
a processor; and
a private memory coupled to the processor, the private memory including a data
structure having a list of thread local storage address size classes wherein each thread
local storage address size class includes a plurality of thread local storage addresses.
9. The apparatus of claim 8, wherein the processor includes a device selected from the
group consisting of microprocessors, programmable logic devices, and microcontrollers.
10. The apparatus of claim 8, further comprising another processor coupled to the
processor.
11. The apparatus of claim 8, further comprising:
a global shared memory coupled to the processor, the global shared memory including
another data structure having another list of thread local storage address size classes.
12. The apparatus of claim 11, wherein the global shared memory can be accessed by a
plurality of processors.
13. The apparatus of claim 8, wherein the private memory can be accessed by a plurality
of processors.

14. The apparatus of claim 8, wherein the data structure includes at least one member selected from the group consisting of singly linked lists, doubly linked lists, binary trees, queues, tables, arrays, sorted arrays, stacks, heaps, and circular linked lists.
15. The apparatus of claim 11, wherein the data structure includes at least one member selected from the group consisting of singly linked lists, doubly linked lists, binary trees, queues, tables, arrays, sorted arrays, stacks, heaps, and circular linked lists.

2019-04-22 14:20:00 -0700